AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method comprising:

removing a work item of a plurality of work items from an enabled expansion bus schedule data structure;

generating a coherency signal independent of said work item utilizing an expansion bus host controller in response to removing said work item from said enabled expansion bus schedule data structure; and

reclaiming ~~resources assigned to~~ said work item by updating a memory, freeing a memory, or reusing a memory ~~whenever~~ after said coherency signal is generated.

2. (Original) The method as set forth in claim 1, wherein said enabled expansion bus schedule data structure comprises an asynchronous schedule including a plurality of queue heads and removing said work item from said enabled expansion bus schedule data structure comprises unlinking a first queue head of said plurality of queue heads from said asynchronous schedule.

3. (Original) The method as set forth in claim 2, wherein said plurality of queue heads includes a second queue head, said second queue head includes a horizontal link pointer to said first queue head, and unlinking said first queue head from said asynchronous schedule comprises modifying said horizontal link pointer of said second queue head.

4. (Original) The method as set forth in claim 2, said method further comprising:

generating a command signal in response to removing said work item from said enabled expansion bus schedule data structure; wherein,

generating a coherency signal utilizing an expansion bus host controller in response to removing said work item from said enabled expansion bus schedule data structure comprises generating a status signal utilizing said expansion bus host controller in response to generating said command signal.

5. (Original) The method as set forth in claim 1, wherein generating a coherency signal utilizing an expansion bus host controller in response to removing said work item from said enabled expansion bus schedule data structure comprises:

traversing said plurality of work items according to a sequence;

storing a copy of a work item within a memory in response to traversing said plurality of work items; and

generating a coherency signal utilizing said copy of said work item.

6. (Original) The method as set forth in claim 5, wherein generating a coherency signal utilizing said copy of said work item comprises:

detecting a removal of said copy of said work item from said memory in response to removing said work item from said enabled expansion bus schedule data structure; and

generating a coherency signal in response to detecting said removal of said copy of said work item from said memory.

7. (Original) The method as set forth in claim 6, wherein detecting a removal of said copy of said work item from said memory in response to removing said work item from said enabled expansion bus schedule data structure comprises detecting a cache flush operation.

8. (Original) The method as set forth in claim 5, wherein generating a coherency signal utilizing said copy of said work item comprises:

identifying an accessible work item of said plurality of work items utilizing said copy of said work item; and

generating a coherency signal in response to traversing beyond said accessible work item in said sequence.

9. (Original) The method as set forth in claim 5, said method further comprising:

executing a transaction on a Universal Serial Bus in response to traversing said plurality of work items.

10. (Original) The method as set forth in claim 1, said method further comprising storing each of said plurality of work items within a memory, wherein reclaiming said work item in response to generating said coherency signal comprises freeing a portion of said memory associated with said work item.

11. (Currently Amended) A computer program product in a recordable-type media that provides instructions which when executed by a machine cause said machine to perform operations comprising:

removing a work item of a plurality of work items from an enabled expansion bus schedule data structure;

generating a coherency signal independent of said work item utilizing an expansion bus host controller in response to removing said work item from said enabled expansion bus schedule data structure; and

reclaiming ~~resources assigned to~~ said work item by updating a memory, freeing a memory, or reusing a memory ~~whenever~~ after said coherency signal is generated.

12. (Previously Presented) The computer program product as set forth in claim 11, wherein said enabled expansion bus schedule data structure comprises an asynchronous schedule including a plurality of queue heads and removing said work item from said enabled expansion bus schedule data structure comprises unlinking a first queue head of said plurality of queue heads from said asynchronous schedule.

13. (Previously Presented) The computer program product as set forth in claim 12, wherein said plurality of queue heads includes a second queue head, said second queue head includes a horizontal link pointer to said first queue head, and unlinking said first queue head from said asynchronous schedule comprises modifying said horizontal link pointer of said second queue head.

14. (Previously Presented) The computer program product as set forth in claim 12, said operations further comprising:

generating a command signal in response to removing said work item from said enabled expansion bus schedule data structure; wherein,

generating a coherency signal utilizing an expansion bus host controller in response to removing said work item from said enabled expansion bus schedule data structure comprises generating a status signal utilizing said expansion bus host controller in response to generating said command signal.

15. (Previously Presented) The computer program product as set forth in claim 11, wherein generating a coherency signal utilizing an expansion bus host controller in response to removing said work item from said enabled expansion bus schedule data structure comprises:

traversing said plurality of work items according to a sequence;

storing a copy of a work item within a memory in response to traversing said plurality of work items; and

generating a coherency signal utilizing said copy of said work item.

16. (Previously Presented) The computer program product as set forth in claim 15, wherein generating a coherency signal utilizing said copy of said work item comprises:

detecting a removal of said copy of said work item from said memory in response to removing said work item from said enabled expansion bus schedule data structure; and

generating a coherency signal in response to detecting said removal of said copy of said work item from said memory.

17. (Previously Presented) The computer program product as set forth in claim 16, wherein detecting a removal of said copy of said work item from said memory in response to removing said work item from said enabled expansion bus schedule data structure comprises detecting a cache flush operation.

18. (Previously Presented) The computer program product as set forth in claim 15, wherein generating a coherency signal utilizing said copy of said work item comprises:

identifying an accessible work item of said plurality of work items utilizing said copy of said work item; and

generating a coherency signal in response to traversing beyond said accessible work item in said sequence.

19. (Previously Presented) The computer program product as set forth in claim 15, said operations further comprising:

executing a transaction on a Universal Serial Bus in response to traversing said plurality of work items.

20. (Previously Presented) The computer program product as set forth in claim 11, said operations further comprising storing each of said plurality of work items within a memory, wherein reclaiming said work item in response to generating said coherency signal comprises freeing a portion of said memory associated with said work item.

21. (Currently Amended) An apparatus comprising:

a command register including a command signal bit to indicate a removal of a work item from an expansion bus schedule data structure including a plurality of work items, wherein the command signal bit is independent of the work item;

a status register including a status signal bit to notify an expansion bus host controller driver that resources assigned to said work item may be reclaimed~~;~~ by updating a memory, freeing a memory, or reusing a memory; and

a microcontroller to process said expansion bus schedule data structure and to modify said status signal bit of said status register in response to said removal of said work item from said expansion bus schedule data structure.

22. (Original) The apparatus as set forth in claim 21, wherein said expansion bus schedule data structure comprises a Universal Serial Bus (USB) asynchronous schedule.

23. (Original) The apparatus as set forth in claim 21, further comprising a cache memory to store a copy of a work item; wherein said microcontroller to process said expansion bus schedule data structure and to modify said status signal bit of said status register comprises:

a microcontroller to traverse said plurality of work items according to a sequence, to store said copy of said work item within said cache memory, and to modify said status signal bit of said status register utilizing said copy of said work item.

24. (Original) The apparatus as set forth in claim 23, wherein said microcontroller to modify said status signal bit of said status register utilizing said copy of said work item comprises:

a microcontroller to modify said status signal bit of said status register in response to a removal of said copy of said work item from said cache memory.

25. (Original) The apparatus as set forth in claim 24, wherein said microcontroller to modify said status signal bit of said status register in response to a removal of said copy of said work item from said cache memory comprises a microcontroller to modify said status signal bit of said status register in response to a cache flush operation.

26. (Original) The apparatus as set forth in claim 23, wherein said microcontroller to modify said status signal bit of said status register utilizing said copy of said work item comprises:

a microcontroller to identify an accessible work item of said plurality of work items utilizing said copy of said work item and to modify said status signal bit of said status register in response to a traversal beyond said accessible work item in said sequence.

27. (Currently Amended) A computer system comprising:

a memory to store an expansion bus schedule data structure including a plurality of work items;

an expansion bus host controller comprising:

a command register including a command signal bit independent of the plurality of work items;

a status register including a status signal bit; and

a microcontroller to process said expansion bus schedule data structure and to modify said status signal bit of said status register in response to a modification of said command signal bit; and

a processor to remove a work item of said plurality of work items from said expansion bus schedule data structure, to modify said command signal bit in response to said removal of said work item from said expansion bus schedule data structure; and to reclaim ~~resources assigned to~~ said work item by updating the memory, freeing the memory, or reusing the memory in response to a modification of said status signal bit.

28. (Original) The computer system as set forth in claim 27, wherein said expansion bus host controller further comprises a cache memory to store a copy of a work item and said microcontroller to process said expansion bus schedule data structure and to modify said status signal bit of said status register comprises:

a microcontroller to traverse said plurality of work items according to a sequence, to store said copy of said work item within said cache memory, and to modify said status signal bit of said status register utilizing said copy of said work item.

29. (Original) The computer system as set forth in claim 28, wherein said microcontroller to modify said status signal bit of said status register utilizing said copy of said work item comprises:

a microcontroller to modify said status signal bit of said status register in response to a removal of said copy of said work item from said cache memory.

30. (Original) The computer system as set forth in claim 28, wherein said microcontroller to modify said status signal bit of said status register utilizing said copy of said work item comprises:

a microcontroller to identify an accessible work item of said plurality of work items utilizing said copy of said work item and to modify said status signal bit of said status register in response to a traversal beyond said accessible work item in said sequence.